Submitted to the AMIA Annual Symposium 2009.
draft material -- please do not cite

Comparing and Evaluating Terminology Services APIs: RxNav, UMLSKS, and LexBIG

Jyotishman Pathak, PhD[1] **Lee Peters, MS**[2] **Christopher G Chute, MD, DrPH**[1] **Olivier Bodenreider, MD, PhD**[2]
[1]Division of Biomedical Statistics and Informatics, Mayo Clinic College of Medicine, Rochester, MN
[2]U.S. National Library of Medicine, National Institutes of Health, Bethesda, MD

Abstract

To facilitate the integration of terminologies into numerous aspects of e-Science, e-Business and e-Government, various terminology services APIs (application programming interfaces) have been developed in the recent past. In this study, we compare and evaluate three publicly available terminology services APIs, RxNav, UMLSKS, and LexBIG, with respect to retrieval of information from one biomedical terminology, RxNorm, common to these services. Our results revealed issues with various aspects of the API implementation and documentation that are currently being addressed.

Introduction

The evolution of terminologies, across the spectrum of detailed nomenclatures and sophisticated classifications, has accelerated dramatically this decade. To facilitate the integration of terminologies into numerous aspects of e-Science, e-Business and e-Government, various terminology services APIs (application programming interfaces) have been developed in the recent past. These APIs, in general, are tuned to (efficiently and effectively) provide a host of functional characteristics ranging from retrieving concept attributes such as definitions and synonyms, to navigating relationships between concepts (e.g., finding sub- or super-concepts of a given concept) and accessing information combinatorially (e.g., list the immediate parent concepts of all concepts that have a term that contains the word `infarction`). Additionally, the APIs provide various degrees of fault resilience, security to prevent unauthorized alteration and/or disruption of content, and the ability to maintain federated linkages between and among components of a single, large terminology or related terminologies with cross-referenced content.

Arguably, terminology services APIs deliver overlapping capabilities and mechanisms for querying the same information, thereby making it important to evaluate the consistency and accuracy of the functionalities provided. The objective of the proposed study is to address this requirement by analyzing three publicly available terminology services APIs, RxNav [1], UMLSKS [2], and LexBIG [3], with respect to retrieval of information from one biomedical terminology, RxNorm, common to these services.

Background

RxNav. RxNav is a browser for RxNorm, the NLM repository of standard names and codes for clinical drugs. RxNav displays links from clinical drugs, both branded and generic, to their active ingredients, drug components and related brand names. RxNav uses Web services API[1] to access the RxNorm data. The API provides various functionalities ranging from searching for a name in the RxNorm data set to get the RxCUIs (Concept Unique Identifiers) to finding relationships between drug products.

UMLSKS. The Unified Medical Language System (UMLS) Knowledge Sources and related lexical programs, developed at the U.S. National Library of Medicine (NLM), provide access to the UMLS. The Metathesaurus, the Semantic Network, and the SPECIALIST lexicon are part of the UMLS, and are typically used by application programs to interpret and refine user queries, to map the user's terms to appropriate controlled vocabularies and classification schemes, to interpret natural language, and to assist in structured data creation. The UMLS Knowledge Source Server (UMLSKS[2]) provides a set of APIs that allow access to the UMLSKS services. The API was developed to support specific queries in order to reduce the total amount of information traveling between the UMLSKS and client applications, and also to provide applications with fine-grained control over the data they wish to receive.

LexBIG. LexBIG is a project that applies the LexGrid vision and technologies to the requirements of the Cancer Biomedical Informatics Grid (caBIG®) community. The goal of the project is to create a vocabulary server built on a well-structured API capable of accessing and distributing vocabularies served via a common information model, namely, the LexGrid Model. This model provides the core representation for all data managed and retrieved through the LexBIG system, and is rich enough to represent vocabularies provided in numerous source formats including the UMLS Rich Release Format (RRF), the Web Ontology Language (OWL), and Open Biomedical Ontologies (OBO). The current implementation of LexBIG provides a robust and flexible tooling for loading, indexing, and managing vocabulary content as well as Java interfaces to various functions including lexical queries, graph representation and hierarchy traversal. It is also compliant with the HL7 Common Terminology Services (CTS I) specification [4].

RxNorm. RxNorm[3], a standardized nomenclature for clinical drugs, is produced by the U.S. National Library of

[1] http://mor.nlm.nih.gov/download/rxnav/RxNormAPI.html
[2] http://umlsks.nlm.nih.gov/DocPortlet/html/dGuide/webservices.html
[3] http://www.nlm.nih.gov/research/umls/rxnorm

RxNorm Term Type	Example
Ingredient (IN)	Fluoxetine
Dose Form (DF)	Oral Solution
Semantic Clinical Drug Component (SCDC)	Fluoxetine 4 MG/ML
Semantic Clinical Drug Form (SCDF)	Fluoxetine Oral Solution
Semantic Clinical Drug (SCD)	Fluoxetine 4 MG/ML Oral Solution
Brand Name (BN)	Prozac
Semantic Branded Drug Component (SBDC)	Fluoxetine 4 MG/ML [Prozac]
Semantic Branded Drug Form (SBDF)	Fluoxetine Oral Solution [Prozac]
Semantic Branded Drug (SBD)	Fluoxetine 4 MG/ML Oral Solution [Prozac]
Branded Pack (BPCK)	Yaz 28 Day Pack
Generic Pack (GPCK)	{31 (Doxycycline 100 MG Oral Tablet)} Pack

Table 1: RxNorm Term Types

Medicine. It contains the names of prescription and many nonprescription formulations approved for human use (primarily in the U.S.). An RxNorm clinical drug name reflects the active ingredients, strengths, and dose form comprising that drug. When any of these elements vary, a new RxNorm drug name is created as a separate concept. Consequently, to distinguish between such drug entities, RxNorm uses "term types" (TTYs) as shown in Table 1. Furthermore, the RxNorm drug entities are related to each other by a well-defined set of named relationships (see Figure 1). For example, ingredient name concepts are related to clinical drug component concepts by the relationships ingredient_of and has_ingredient. Finally, RxNorm also contains a list of identifiers from other vocabularies that appear as concept attributes (see Table 2).

Materials

The following materials were used in this study:

- RxNav API 1.0 released in October, 2008 and accessible via: http://mor.nlm.nih.gov/download/rxnav/RxNormAPI.html.
- UMLSKS API 5.2 released in July, 2005 and accessible via: http://umlsks.nlm.nih.gov.
- LexBIG API 2.3 released in October, 2008 and accessible via: https://gforge.nci.nih.gov/projects/lexevs.
- RxNorm November 17, 2008 Full Update Release data that is consistent with the 2008AB version of the UMLS, and accessible via: http://download.nlm.nih.gov/umls/kss/rxnorm/RxNorm_full_11172008.zip. This dataset included 4,112 ingredients, 100 dose forms, 13,923 clinical drug components, 8,180 clinical drug forms, 18,228 clinical drugs, 10,029 brand names, 14,154 branded drug components, 11,643 branded drug forms, 14,891 branded drugs, 288 branded packs, and 224 generic packs. Furthermore, the dataset had over 500,000 relationships between these RxNorm entities.

RxNorm idType	Identifier Name
AMPID	Alchemy Marketed Product Identifier
GCN	Generic Code Number
GFC	Generic Formula Code
GPPC	Generic Product Packing Code
GS	Gold Standard Alchemy Identifier
LISTING_SEQ_NO	FDA Identification Number
MMSL_CODE	Multum Identifier
NDC	National Drug Code
SNOMEDCT	SNOMEDCT Identifier
SPL	Standard Product Label
UMLSCUI	UMLS Concept Unique Identifier
VUID	Veterans Health Administration Unique Identifier

Table 2: RxNorm Vocabulary Identifiers

Methods

For this study, we established a list of queries (see Figure 2) that cover a wide spectrum of terminology services functionalities such as finding RxNorm concepts by their name, or navigating different types of relationships based on the current implementation of the RxNav API. In order to query for relationships between various RxNorm drug entities, a list of preferred paths among categories of entities in RxNorm was developed (see Table 3 for a snapshot and RxNav API documentation for details). For example, given a brand name *Tylenol PM* (RxCUI=220581), one can retrieve the ingredients *Acetaminophen* (RxCUI=161) and *Diphenhydramine* (RxCUI=3498) by traversing the direct path between BN and IN via the relationship tradename_of. On the other hand, to retrieve the clinical drugs *Acetaminophen 33.3 MG/ML* (RxCUI=328877) and *Diphenhydramine 1.67 MG/ML* (RxCUI=333781) for *Tylenol PM*, one has to traverse the indirect path between BN and SCD via the relationships ingredient_of and tradename_of.

Based on these API calls, training and test data was generated from the RxNorm dataset to verify and evaluate the implementation of the functionalities, respectively. Furthermore, to facilitate the exchange of queries and analysis of the result set, an XML Schema was established that was loosely based on the SOAP envelope of the RxNav Web Service API (refer to the WSDL schema from: http://mor.nlm.nih.gov/download/rxnav/RxNormDBService.wsdl). Finally, the output results generated by UMLSKS and LexBIG were evaluated against the resultset from RxNav.

Results

Table 4 summarizes the results for UMLSKS and LexBIG APIs compared with the RxNav API. The first column indicates the type of query evaluated (as listed in Figure 2) along with the number of queries executed (in the test data) in the second column. The third and fourth columns refers to the number of queries that differed in the results from UMLSKS and LexBIG APIs, respectively, compared to the results from the RxNav API for the test data.

Discussion

Evaluation. In all the cases, many differences were observed between the results returned by the individual APIs.

For **findRxcuiByString**, 19 differences were observed

Start TTY	End TTY	Preferred Path
BN	IN	BN ⇒ IN
BN	SCD	BN ⇒ SBD ⇒ SCD
IN	SCDC	IN ⇒ SCDC
IN	DF	IN ⇒ SCDC ⇒ SCD ⇒ DF
SBD	SBDF	SBD ⇒ SBDF
BPCK	SCD	BPCK ⇒ SBD ⇒ SCD "or" BPCK ⇒ SCD

Table 3: Example RxNorm Term Types and Preferred Paths

between the RxNav and UMLSKS resultsets, of which 17 were caused by slight differences between the two datasets used for querying RxNav and UMLSKS. Specifically, even though RxNorm November 17, 2008 Full Update Release data was aligned with the 2008AB version of the UMLS (used for querying the UMLSKS), some RxNorm concepts were missing from the UMLS release. The other 2 differences were not associated with the dataset alignment issues: the first difference occurred in searching for "*psyllium husk*". UMLSKS returned two RxCUIs, 104129 ("*psyllium husk*") and 8928 ("*psyllium*"), although the second RxCUI was not found by RxNav. Investigating further, we realized that UMLSKS found 8298 because "*psyllium husk*" is a synonym from the NCI Thesaurus for "*psyllium*", and the RxNorm dataset does not contain terms from the NCI Thesaurus. The second difference occurred because of different exact match rules between the two APIs—the search for "Senna Lax" yielded RxCUIs 219861 ("*Senna Lax*") and 219864 ("*Sennalax*") in UMLSKS, while RxNav only found 219861 ("*Senna Lax*"). On the other hand, for LexBIG, out of 6 differences, 3 were due to search strings not found by the LexBIG (and found by RxNav), 2 were due to LexBIG returning obsolete RxNorm concepts, and finally, LexBIG returned one RxCUI without a RxNorm term.

For **findRxcuiById**, there were 124 differences between UMLSKS and RxNav resultsets, of which 100 were due to the lack of existing capability in UMLSKS to search by idType=NDC. The remaining 24 differences were a result of imperfect alignment between the RxNorm datasets as elucidated above. For LexBIG, 66 differences were observed with the RxNav resultsets. In particular, for idType=GCN, LexBIG returned 29 RxCUIs which belonged to obsolete RxNorm data. Additionally, one of the returned RxCUI had ET as its term type, which is invalid (see Table 1). Similar observations were made for idType=LISTING_SEQ_NO, where one of the returned RxCUI pointed to obsolete data, and another RxCUI did not have an RxNorm term (this was also true for idType=SNOMEDCT). Interestingly, for idType=NDC, LexBIG not only returned all the expected RxCUIs, but also found 33 NDC identifiers that had more than one RxCUI. The RxNav API, on the other hand, returned only one RxCUI per identifier; a behavior that can be attributed to the RxNav interface selecting one identifier for highlighting purposes. Furthermore, for this evaluation, the results from idType=MMSL_Code were excluded since RxNav did not find any matches, due to the fact that the identifiers were not in the format required by the API.

For **getNDCs**, 4 differences were observed between the UMLSKS and RxNav results which were due to imperfect alignment between the RxNorm datasets. On the other hand, LexBIG and RxNav results had no differences.

Query Type	# of Queries	UMLSKS Differences	LexBIG Differences
findRxcuiByString	820	19	6
findRxcuiById	1100	124	66
getNDCs	100	4	0
getRxConceptProperties	102	5	0
getProprietaryInformation	100	4	100
getRelatedByRelationship	1060	58	66
getRelatedByType	820	40	80
getAllRelatedInfo	100	18	37

Table 4: Query Result Comparison with the RxNav API

For **getRxConceptProperties**, 5 differences were observed between the UMLSKS and RxNav results, and all were a result of imperfect alignment between the RxNorm datasets. On the other hand, there were no differences between the LexBIG and RxNav results.

For **getProprietaryInformation**, 4 differences were observed between the UMLSKS and RxNav results, and all were a result of imperfect alignment between the RxNorm datasets. On the other hand, LexBIG could not return results for any of the queries since such information is not captured by RxNorm (RRF) loader for LexBIG.

For **getRelatedByRelationship**, 58 differences were observed between the UMLSKS and RxNav results, and all were a result of imperfect alignment between the RxNorm datasets. Whereas, for LexBIG, 66 differences were observed, and all of them were due to the result of LexBIG returning obsolete RxNorm concepts.

For **getRelatedByType**, 40 differences were observed between the UMLSKS and RxNav results, of which 38 were due to the imperfect alignment between the RxNorm datasets. The other 2 differences were observed when UMLSKS retrieved the desired results, but RxNav failed due to issues in processing BPCKs and GPCKs. For LexBIG, we observed 80 differences with the RxNav results, and a bulk of which (61) were due to LexBIG returning obsolete data. Additionally, 16 differences were observed when the target type was IN, and LexBIG results were missing the precise ingredients (PIN), and one difference occurred (RxCUI=236216, endTTY=SCD) where the LexBIG results did not return the SCDs associated with the PIN. Similar to UMLSKS, the other 2 differences were observed due to issues in RxNav processing of BPCKs and GPCKs.

For **getAllRelatedInfo**, 18 differences were observed between the UMLSKS and RxNav results, of which 17 were a result of imperfectly aligned RxNorm datasets. One difference occurred (RxCUI=494944) where UMLSKS results did not find a DF concept. On the other hand, 37 results were different in LexBIG compared to RxNav, of which 19 differences were due to LexBIG returning obsolete RxNorm concepts. For the remainder, 16 differences were observed when the target type was IN, and LexBIG results were missing the precise ingredients (PIN), one difference occurred (RxCUI=2625) where LexBIG did not return the SCD associated with the PIN, and finally, another difference occurred (RxCUI=494944) where LexBIG did not find a DF concept.

Practical Implications. The implications of this study were manifold. As illustrated in our evaluation, leaving aside minor nuances, all the three APIs were functionally similar in terms of information retrieval, and differences in resultsets were primarily due to issues in dataset alignment and content loading.

In particular, for LexBIG, major differences in results were due to returning obsolete RxNorm concepts for the queries performed. The information about obsolete concepts, although present in the RxNorm dataset, is not captured by the Rich Release Format (RRF) loader in LexBIG. Similarly, all the concept information associated with a concept for the specified sources is not captured by the RRF loader in a consistent way. For instance, the information about the mapping between a RxNorm concept (e.g., RxCui=161 ("*Acetaminophen*")) to a concept in another vocabulary (e.g., id=5005 in MMSL) is missing in many cases, and as a consequence, LexBIG could not return results for the **getProprietaryInformation** query. We have created a entry for this issue in the LexBIG bugtracker, and we expect it to be fixed in the next LexBIG release (June, 2009). However, at the same time, the LexBIG API was highly performant and provided various convenience methods to uniformly query its underlying common information LexGrid model.

For UMLSKS, the differences observed in resultsets were mainly due to one reason: imperfect alignment of 2008AB release of UMLS with the RxNorm November 17, 2008 Full Update Release. While addressing this issue is beyond the scope of this work, we realized that typically the RxNorm dataset is submitted to the UMLS maintainers a few months before the UMLS scheduled release date. Consequently, by the time the UMLS Metathesaurus is made publicly available, the RxNorm dataset would have evolved due to addition of new drugs or elimination of obsolete ones, thereby causing the RxNorm dataset to have new data (without UMLSCUI information) as well as eliminated old data. The UMLSKS API also did not explore all the features of the RxNorm dataset. For example, it was not possible to search for RxCUIs using NDC identifiers. Additionally, during the training set analysis, it was discovered that UMLSKS incorrectly returned concepts for **findRxcuiByString** when the search string had more than 30 characters. For example, when executing **findRxcuiByString** for the string "*Benztropine Injectable Solution*", UMLSKS returned RxCUIs 371036 ("*Benztropine Injectable Solution*") and 92198 ("*Benztropine Injectable Solution [Cogentin]*"), of which the latter is incorrect. After notifying the UMLSKS maintainers about this issue, a problem in the string matching algorithm was discovered, and subsequently fixed before analyzing our test data. One of the search features in UMLSKS that could benefit RxNav is the removal of special characters from the search string for exact matches. For example, searching for "Senna Lax" yielded RxCUIs 219861 ("*Senna Lax*") and 219864 ("*Sennalax*") in UMLSKS, but RxNav only retrieved 219861. Note that, LexBIG already implements such a feature.

For RxNav, in addition to issues with exact match string searching, we discovered problems involving GPCK and BPCK processing where the preferred path was not traversed. For example, when executing **getRelatedByType** with RxCUI=750119 ("*Tirosint 0.013 56 Day Pack*") and TTY=SCDC, RxNav returned no results. The correct result is the concept "*Thyroxine 0.013 MG*" (RxCUI=728558) which was returned by both UMLSKS and LexBIG. This issue will be fixed in the next release of RxNav API. Furthermore, for **findRxcuiById**, RxNav returned only one RxCUI per identifier (such as NDC), although the dataset contained more than one RxCUI in some cases. This issue has also been brought to the attention of the RxNav development team. Furthermore, we observed that documentation for few functionalities implemented by RxNav was sparse, and required significant enhancements. In particular, the documentation about the RxNorm graph theory as well as relationship mappings between RxNorm entities need to be improved.

Limitations. The study only evaluated retrieval of information from one biomedical terminology: RxNorm. In the future, we plan to expand our investigation by incorporating more terminology sources, although arguably, APIs such as RxNav, developed specifically for a particular terminology, will not be applicable. Furthermore, we intend to include additional publicly available terminology services APIs such as Apelon DTS [5] in our study. Another aspect of our investigation which requires further evaluation is running performance benchmarks, and analyzing various degrees of fault resilience and load balancing capabilities of the APIs.

Conclusion

In this study, we experimented with three publicly available terminology services APIs to query a clinical drug terminology, RxNorm, and highlighted various issues. Our investigation, a first of its kind, contributed to provide a methodological model in comparing and evaluating terminology services APIs developed by different organizations.

Acknowledgment. This research is supported in part by NHGRI grant U01HG04599 (eMERGE), and the Intramural Research Program of the National Institutes of Health (NIH), National Library of Medicine (NLM).

References

[1] Peters L, Bodenreider O. Using the RxNorm Web Services API for Quality Assurance Purposes. In: AMIA Annual Symposium. AMIA Proceedings; 2008. p. 591–595.

[2] Bangalore A, Thorn KE, Tilley C, Peters L. The UMLS Knowledge Source Server: An Object Model for Delivering UMLS Data. In: AMIA Annual Symposium. AMIA Proceedings; 2003. p. 51–55.

[3] Pathak J, Solbrig HR, Buntrock JD, Johnson TM, Chute CG. LexGrid: A Framework for Representing, Storing, and Querying Biomedical Terminologies from Simple to Sublime. Journal of the American Medical Informatics Association. 2009;16(3).

[4] HL7 Common Terminology Services (CTS) Specification;. Available from: http://www.eclipse.org/ohf/components/cts.

[5] Apelon Distributed Terminology System;. Available from: http://www.apelon.com/products/dts.htm.

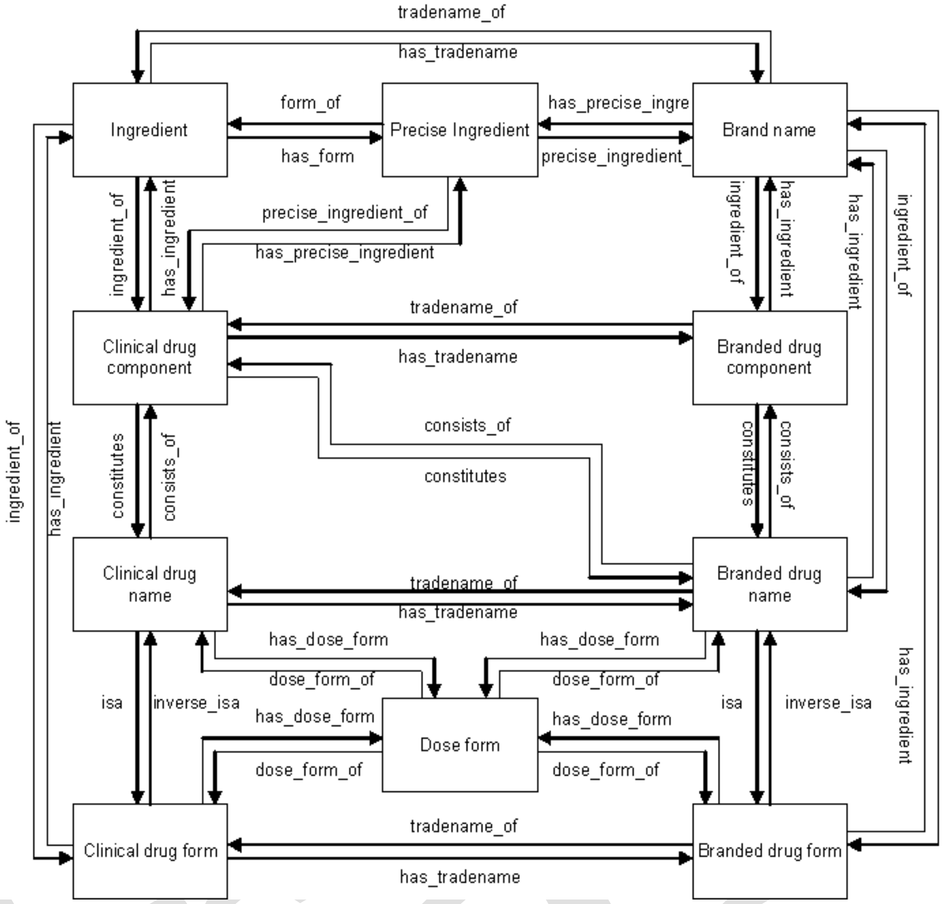

Figure 1: Relationships between RxNorm Drug Entities

- **findRxcuiByString(*searchString*)** Search for a name in the RxNorm data set and return the RxCUIs of any concepts which have that name as an RxNorm term or as a synonym of an RxNorm term.
- **findRxcuiById(*idType,id*)** Search for an identifier from another vocabulary and return the RxCUIs of any concepts which have an RxNorm term as a synonym or have that identifier as an attribute.
- **getNDCs(*rxcui*)** Get the National Drug Codes (NDCs) for the RxNorm concept.
- **getAllRelatedInfo(*rxcui*)** Get all the related RxNorm concepts for a given RxNorm identifier.
- **getRxConceptProperties(*rxcui*)** Get the RxNorm Concept properties.
- **getRelatedByType(*rxcui,typeList*)** Get the related RxNorm identifiers of an RxNorm concept specified by one or more term types.
- **getRelatedByRelationship(*rxcui,relaList*)** Get the related RxNorm identifiers of an RxNorm concept specified by a relational attribute list.
- **getProprietaryInformation(*rxcui,sourcelist,proxyTicket*)** Get the concept information associated with the concept for the specified sources. The user must have a valid UMLS license and be able to access the UMLSKS authority service to obtain proxy tickets to use this function.

Figure 2: Query Functionalities Implemented in RxNav, UMLSKS and LexBIG